BASE DE DONNÉES – TD4 BONUS

PL / SQL

**LÊ NGUYỄN MINH HƯƠNG**

# 2. PROCÉDURE STOCKÉE :

## 3/ Écrivez une fonction qui calcule la moyenne d’un étudiant. Cette fonction doit générer une exception si jamais l’élève n’a pas de note.

create or replace function calculerMoyenne (n\_etu number) return number as

e exception;

nb number;

res number(4,2);

somme number;

nb\_coef number;

cursor cs\_moyenne is

select note, coeff\_mat

from etudiants e, matieres m, notes n

where e.num\_etu = n.num\_etu

and n.code\_mat = m.code\_mat

and e.num\_etu = n\_etu;

begin

select count(\*) into nb

from etudiants e, notes n

where e.num\_etu = n.num\_etu

and e.num\_etu = n\_etu;

if nb = 0 then raise e;

else

somme:=0;

nb\_coef:=0;

for l in cs\_moyenne

loop

somme := somme + l.note\*l.coeff\_mat ;

nb\_coef := nb\_coef + l.coeff\_mat ;

end loop;

res:=somme/nb\_coef;

return res;

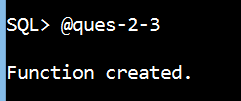
end if;

exception

when e then

return 0;

end calculerMoyenne;



*// exécuter la fonction*set serveroutput on;

accept numero\_etu number prompt 'Entrez le numero de l''etudiant : ';

declare

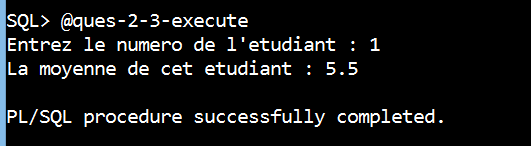
moyen number;

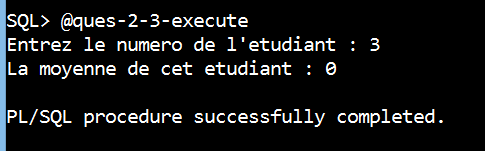
begin

moyen:=calculerMoyenne(&numero\_etu);

dbms\_output.put\_line('La moyenne de cet etudiant : ' || moyen);

end;





## 4/ Écrivez une procédure qui fait passer en année supérieure les étudiants ayant plus de 11 de moyenne et supprimant tous les étudiants ayant moins de 8 de moyenne ou n’ayant aucune note.

create or replace procedure passerEtudiant as

moyenne number;

cursor cs\_etu is

select num\_etu, annee\_etu

from etudiants;

begin

for l in cs\_etu

loop

moyenne := calculerMoyenne(l.num\_etu);

if moyenne > 11 then

update etudiants set annee\_etu = annee\_etu + 1 where num\_etu = l.num\_etu;

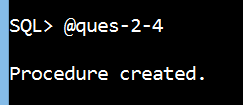
elsif moyenne < 8 or moyenne = 0 then

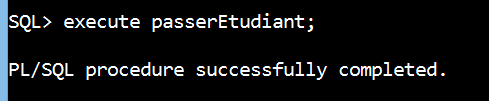
delete from etudiants where num\_etu = l.num\_etu;

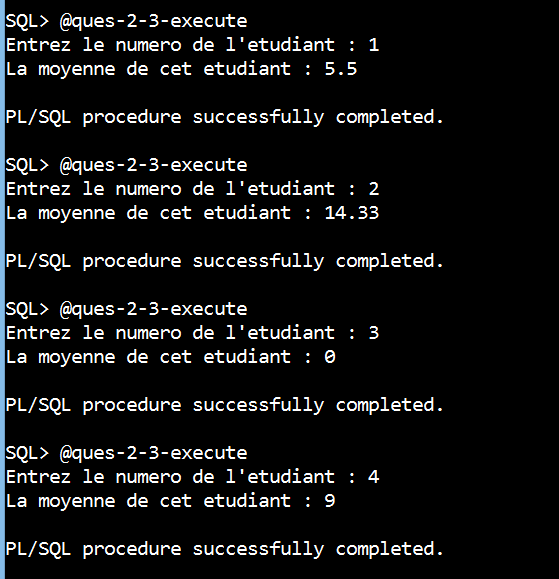
end if;

end loop;

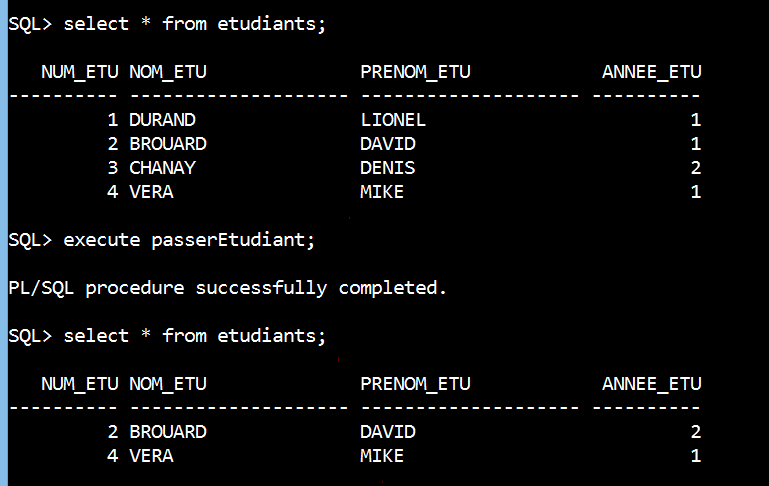
end;







Calculer la moyenne de chaque étudiant pour tester



# 3. DÉCLENCHEUR :

## 1/ Créez un déclencheur qui s’assure qu’un étudiant ne peut pas sauter de classe ni passer à une classe inférieure.

set serveroutput on;

create or replace trigger trg\_annee

after update on etudiants

for each row

begin

if updating then

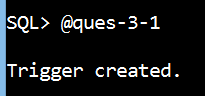
if (:new.annee\_etu - :old.annee\_etu > 1) or (:new.annee\_etu - :old.annee\_etu < 0) then

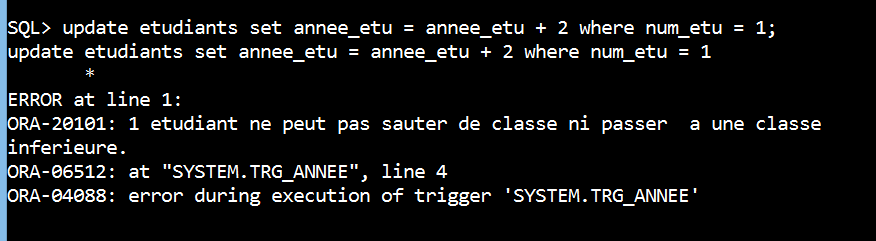
raise\_application\_error(-20101, '1 etudiant ne peut pas sauter de classe ni passer a une classe inferieure.');

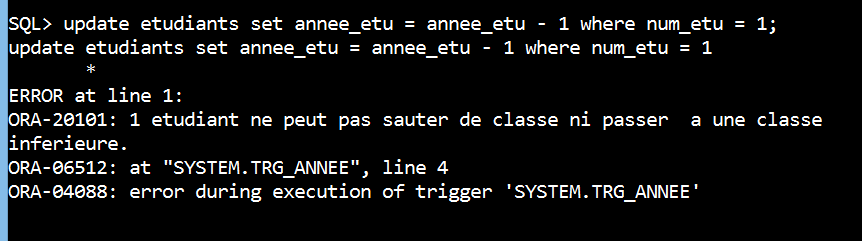
end if;

end if;

end trg\_annee;







## 2/ Créez un déclencheur qui permet de conserver la liste des anciens élèves.

create or replace trigger trg\_conserver

after delete on etudiants

for each row

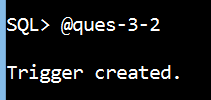
begin

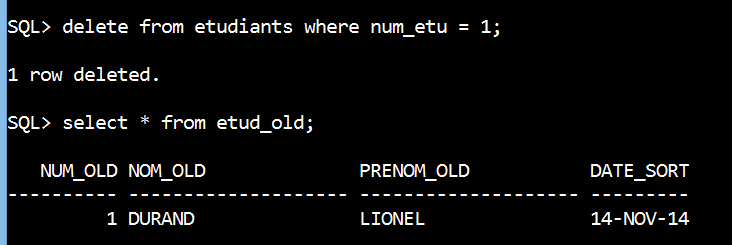
if deleting then

insert into etud\_old values (:old.num\_etu, :old.nom\_etu, :old.prenom\_etu, SYSDATE) ;

end if;

end trg\_conserver;





Soit on peut modifier légèrement le script creation1.sql pour que l’année s’écrit avec 4 lettres :

CREATE TABLE ETUD\_OLD (

NUM\_OLD NUMBER PRIMARY KEY,

NOM\_OLD VARCHAR2(20),

PRENOM\_OLD VARCHAR2(20),

ANNEE\_SORTIE DATE

);

ALTER SESSION SET NLS\_DATE\_FORMAT = 'DD-MM-YYYY';

Autre résultat :

